Protocol for a Systematic Literature Review on Adaptative Middleware Support for IoT and CPS

Mahyar T Moghaddam, Eric Rutten, Guillaume Giraud

Protocol for a Systematic Literature Review on Adaptive Middleware Support for IoT and CPS

Mahyar T. Moghaddam *
Univ. Grenoble Alpes, Inria, CNRS, LIG
F-38000 Grenoble France
mahyar.tourchi-moghaddam@inria.fr

Eric Rutten
Univ. Grenoble Alpes, Inria, CNRS, LIG
F-38000 Grenoble France
eric.rutten@inria.fr

Guillaume Giraud
RTE
Paris, France F-92073
guillaume-np.giraud@rte-france.com

Abstract

This protocol defines the procedure to conduct a systematic literature review on adaptive middleware support for the Internet of Things (IoT) and Cyber-physical Systems (CPS). The mentioned concepts deal with smart interactive objects which provide a set of services, but they look into the problem from various perspectives. We especially look into middleware design decisions for reactive/proactive adaptations. Following a systematic literature review (SLR) in the selection procedure, we selected 62 papers among 4,274 candidate studies. To this end, we applied the classification and extraction framework to select and analyze the most influential domain-related information. In addition to the academic database, we took advantage of the use-cases provided by our industrial partners within the CPS4EU [2] project. This document clarifies the primary studies' selection process. The analysis of the studies, discussion, and solution proposals will be presented separately in a journal article.

***K*eywords** Internet of Things · Industrial Internet of Things · Cyber-physical systems · Pervasive · Middleware · Self-adaptation · Systematic Literature Review

1 Introduction

New architectural styles, tools, and techniques can shape the future use of IoT/CPS. The appropriateness of such methods depends on the unification of heterogeneous things. However, the quality issues can guide the overall system towards success or failure. CPS, IoT, and pervasive systems combine aspects of the physical and digital worlds. While IoT emphasizes connectivity, CPS underlines the embedded aspect, and pervasive systems highlight the ubiquitous computing.

IoT/CPS are made up of sensing, communication, processing, middleware, and actuation elements [1]. The sensing elements get real-life data by embedded devices that make use of sensors. The communication elements provide the mechanism and protocols to transmit sensed data to processing and storage components. The processing elements analyze the data to plan for actuation. The actuation elements provide the services that the IoT/CPS infrastructure aims to provide. Middleware that runs in processing components facilitates the communication between heterogeneous sensing and actuating components using a set of programming abstractions. Middleware is a software layer between

*Corresponding author.

[2] CPS4EU is a three years project funded by the H2020-ECSEL-2018-IA. The project develops four vital IoT technologies, namely computing, connectivity, sensing, and cooperative systems. It incorporates those IoT technologies through pre-integrated architectures and design tools. It instantiates the architectures in dedicated use-cases from a strategic application viewpoint for automotive, smart grid, and industrial automation. https://cps4eu.eu

the physical and application layers, which provides a set of programming abstractions to facilitate the integration and communication of heterogeneous components pervasively.

The IoT/CPS characteristics may require various levels of processing elements' distribution, localization, and collaboration [2].

- *Distribution*: this aspect specifies whether data analysis software ought to be deployed on a single node or several nodes distributed across the IoT system.
- *Localization*: depending on data size and required analysis, complexity, processing can be executed locally or remotely. Here is the point in which centralized cloud and distributed edge and fog concepts become relevant.
- *Collaboration*: the processing components may interact to form and empower IoT services. This collaboration may appear as a level of information sharing, coordinated analysis, and/or planning or synchronized actuation [3].

One crucial aspect that comprises all IoT/CPS components is the ability to be adapted based on both the system itself and its environmental situation. Adaptation techniques can also guarantee the dynamic nature of collaborative systems. Adaptation typically addresses control elements that interact with the environment to provide a service. Feedback control loops model such interactions. Feedback loops [4, 3] can support both design-time and run-time adaptation. In this regard, the middleware components should facilitate and manage the adaptation, potentially via interacting loops that capture and incorporate contextual information at various layers of the system.

The goal of this paper is to identify, classify, and propose a set of state-of-the-art based architectures, tools, and techniques that are potentially suitable to model (self-) adaptive IoT/CPS. More specifically, we discuss various challenges tied up with middleware support for (self-) adaptive IoT/CPS and propose adequate solutions. We are interested in interacting control loop mechanisms capable of supporting hard/soft real-time functionality of IoT/CPS and satisfying crucial non-functional requirements.

A body of knowledge exists in the scientific literature about IoT/CPS middleware support. However, those studies are scattered across different independent research areas, such as software engineering, embedded systems, and networking. Therefore, a survey that classifies and compares the various approaches and methods for understanding the *IoT/CPS adaptation objectives, control mechanisms* and *middleware support* is still missing. The study identifies current characteristics, challenges and publication trends, and research gaps concerning (self-) adaptive middleware support for IoT/CPS approach. This study's audience is both research and industry communities interested in improving their knowledge and selecting suitable methods to design and develop their middleware for IoT/CPS.

This protocol includes the main elements of a systematic study that namely are: *i)* motivation, *ii)* research questions, *iii)* a search string to discover relevant studies, *iv)* inclusion and exclusion criteria to determine relevant studies among the initial set, *v)* relevant data sources, *vi)* research team and external reviewers, *vii)* the primary studies distribution.

2 Motivation

This section discusses the motivation for handling our research and its potential scientific value. To this end, an extensive search has been carried out in Sub-section *2.1* to discover the related reviews. By comparing this research with already conducted systematic studies in the field, the current knowledge gap can be discovered. Sub-section *2.2* gives concise reasoning upon the necessity for a systematic review of (self-) adaptive middleware support for CPS and IoT.

2.1 Existing Related Literature Reviews

In order to uncover previous systematic literature review (SLR) and systematic mapping study (SMS) related to this research topic, we performed a search on relevant databases [3] using the following string. To include all related articles, we applied the string on title, abstract, and keywords.

("mapping study" OR "literature review" OR SLR OR SMS) AND (IoT OR "Internet of Things" OR IIoT OR CPS OR "cyber-physical" OR "cyber physical" OR cyberphysical OR pervasive) AND (middleware OR "middle-ware" OR "middle ware") AND ("self-adapt" OR "self adapt*" OR "self*" OR "adapt*" OR autonomic)*

The string aims at discovering any systematic review on (self-) adaptive middleware support for IoT/CPS. We included all peer-reviewed systematic reviews and mapping studies that discuss any architectural, technical, or practical aspects

[3] *ACM, IEEE Xplore, SpringerLink, Web of Science, Scopus, Wiley, and ScienceDirect*

of (self-) adaptive middleware support for CPS/IoT. Short articles and papers which focus on any CPS/IoT aspect other than middleware support were excluded.

We analyzed the search results, but we did not find any systematic study on the topic. However, six slightly related studies with different scopes have been chosen to be compared with our research. These studies were selected since they address some aspects of (self-) adaptation or middleware design in IoT or CPS. Table 1 shows the existing systematic studies, their focus, and the associated quality assessment (based on [5, 6]). We calculated the total score of each study [7, 5] by summing up the answer to each specific question *Q1-Q4* (Yes(Y)=1, Partly(P)=0.5, No(N)=0):

- **Q1)** Are the systematic study's inclusion and exclusion criteria described appropriately?
- **Q2)** Is the literature search likely to have covered all relevant studies?
- **Q3)** Did the authors assess the quality and validity of the included studies?
- **Q4)** Were the basic concepts and gathered data adequately described?

Table 1: Existing systematic studies on CPS and IoT middleware support.

Study	*Focus*	*Year*	*Q1*	*Q2*	*Q3*	*Q4*	*Total Score*
1. Self-Adaptation for Cyber-Physical Systems:A Systematic Literature Review [8]	Architectural self-adaptation in CPS	2016	Y	Y	Y	Y	4
2. Architecting cloud-enabled systems: a systematic survey of challenges and solutions [9]	Cloud-based software systems architecture with a focus on middleware services	2016	Y	Y	Y	Y	4
3. Control-Theoretical Software Adaptation: A Systematic Literature Review [10]	control-theoretical software adaptation mechanisms	2017	Y	Y	Y	Y	4
4. Fog Computing Applications in Smart Cities: A Systematic Survey [11]	Various solutions provided by Fog computing in smart cities context	2020	Y	P	P	Y	3
5. A comprehensive and systematic review of the load balancing mechanisms in the Internet of Things [12]	analyzing and examining load balancing remarkable methods	2019	N	P	P	Y	2
6. Service-Oriented Middleware Architectures for Cyber-Physical Systems [13]	Review on CPS midlware and presenting a conceptual middleware design	2012	P	P	P	Y	2.5

Research 1 [8] studies state-of-the-art approaches to handle self-adaptation in CPS at the architectural level. The paper follows a transparent methodology to present a reference three-layer adaptation model. The most relevant studies are included, and the results are well described. The report analyzes the existing approaches to self-adaptation architecture in CPS to better understand state of the art and propose various solutions. While the paper considers the use of MAPE-K (Monitoring, Analysis, Planning, Execution, and Knowledge) loop for CPS self-adaptation, it does not investigate multiple interacting loops. Furthermore, the authors do not focus on analyzing novel middleware technologies. Instead, our study widens the scope to IoT and pervasive systems and proposes a set of middleware solutions for distributed systems.

Study 2 [9] respects all steps of a systematic review from inclusion/exclusion criteria to data analysis. The paper identifies 44 unique categories of challenges and associated solutions for architecting cloud-based software systems. The authors suggest that many primary studies focus on middleware services to achieve scalability, performance, response time, and efficient resource optimization. The challenge has been observed in various domains, form pervasive embedded systems and enterprise applications to smart IoT devices. While the paper addresses the use of Domain-Specific Languages in modeling secure CPS, it ignores suggesting other solutions such as service-oriented approaches. Our study characterizes device edge and fog as well, which can enhance the IoT/CPS quality.

Study 3 [10] thoroughly followed the systematic reviews' steps and protocols. This paper investigates software adaptation by modifying the software rather than the resource allocated to its execution. This paper mainly focuses on control-theoretical software adaptation and control mechanisms. The paper investigates control loops, but it ignores other IoT/CPS middleware aspects such as requirements, tools, and techniques.

Study 4 [11] follows the systematic mapping study method to obtain an overview of the existing related research literature on fog and cloud-based smart cities applications. The paper presents an analytical comparison of related works, the trends, and future research directions on Fog computing. Our study's advantage over [11] is that we build our middleware modeling styles on top of the perceived knowledge from the reviewed literature and industrial use-cases.

Study 5 [12] respects the systematic review process, such as explaining the research questions and (partially) addressing the inclusion and exclusion criteria. The paper investigates optimizing the usage of IoT networks by providing solutions for scalability, routing, reliability, security, energy conservation, network lifetime, congestion, heterogeneity, and quality of service (QoS). The authors deal with QoS issues such as latency and data packets loss using load balancing concept by distributing loads among different routes. Our study, instead, deals with the QoS int both system and middleware architectural levels.

Study 6 [13] first present a systematic literature survey of research outputs in CPS middleware designs *i)* to present the state-of-the-art and *ii)* to bring out some research focus on the issue. The authors further propose an early conceptual middleware designed with a service-oriented viewpoint to support CPS applications. Our study includes all architectural styles and patterns that can be useful for research and industry.

2.2 Need for an SLR on IoT/CPS modeling

The need for CPS modeling is augmented by the advent of IoT, where the relationship between physical and virtual worlds plays a fundamental role. This research complements the existing studies regarding the adaptation in IoT/CPS middleware support by introducing a literature-based classification of the objectives, decision methods, and tools. Although the IoT/CPS research started by concepts appeared more than two decades ago, the research and industry communities are still in their progress to define its different aspects effectively. To discover the impact of existing literature on adaptive IoT/CPS middleware support, we identify, describe, and classify various concepts and techniques used to engineer industry-oriented systems to help practitioners choose the best modeling tool.

3 Research Implementation

This study has been carried out according to systematic reviews guidelines provided in [14, 15, 5, 6]. In this regard, we formulized our perspective by defining the purpose, issue, object, viewpoint issues ([16]).

Purpose: to provide a deep understanding of middleware support for (self-) adaptive IoT/CPS

Issue: by identifying, classifying, and analyzing different objectives, decision methods, and tools

Object: based on existing IoT/CPS adaptation approaches

Viewpoint: from the research and industry viewpoints.

Such an approach comes as the primary aim of this study since there is no proper overview of (self-) adaptive IoT/CPS middleware support, which considers adaptive infrastructure and environment interaction, adaptation decision methods, and tool support with an industrial orientation. As shown in Figure 1, the overall process can be divided into three main phases ([15], [17]): *planning, conducting, and documenting*.

External Review. The protocol and final reports will be sent to external experts for independent review to mitigate potential threats to validity and biases. As shown in Table 2, we received recommendations from the following scientists:

- *Software Engineering and SLR Expert*. Patrizio Pelliccione is an associate professor at the University of L'Aquila / Chalmers University of Technology with a research focus on software engineering, robotics, and autonomous systems. He has extensive experience in industrial empirical studies and systematic reviews. He is contacted to provide feedback about the research's general structure, primary studies, and data extraction criteria.
- *Software Architecture and SLR Expert*. Ivano Malavolta is an assistant professor at the Vrije Universiteit Amsterdam, who works on software architecture, mobile software, and robotics software. He has published several systematic studies on various software engineering fields, such as microservices, self-adaptation, and collaborative modeling. He advises architectures, methods, and tools that are well known and of interest for the IoT/CPS community. He also confirms the search string coverage and inclusion/exclusion criteria.

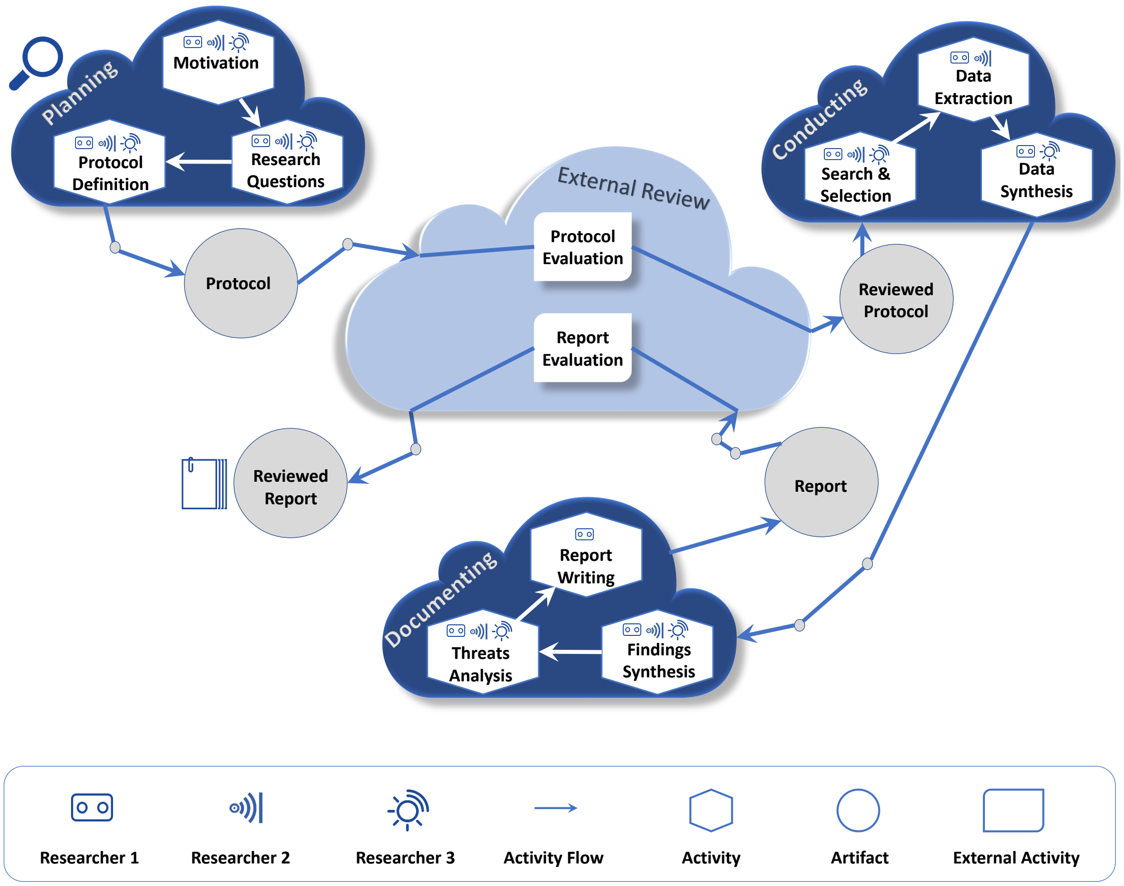

Figure 1: Overall review process overview.

Table 2: External reviewers.

Software Engineering and SLR Expert	Software Architecture and SLR Expert
Patrizio Pelliccione	Ivano Malavolta

External reviewers had a period of two weeks to provide their feedback about the proposed artifacts. Their comments were useful to enhance the quality of the study.

3.1 Planning

Planning aims at: *i)* establishing the need for performing a literature review on (self-) adaptive middleware support for IoT/CPS (see Section 2), *ii)* identifying the main research questions (see Section 3.2.1), and *iii)* defining the protocol (see the remainder of this document) [5]. The planning phase is structured through the existing document.

3.2 Conducting

This step includes the search and selection strategies and data extraction and synthesis as follows.

3.2.1 Search Strategy

To achieve the research goal, we arranged for a set of questions along with their rationale (Table 3). The classification resulting from our investigation on the research questions will provide a solid foundation for a thorough identification and comparison of existing and future solutions for (self-) adaptive middleware for IoT/CPS. This contribution is useful for both researchers and practitioners willing to further contribute to new IoT/CPS modeling and development approaches or better understand or refine existing practices. The research questions listed in Table 3 will drive the

Table 3: Research questions and the respective rationale.

#	*Questions*	*Sub-questions*	*Rationale*
RQ1	What are the objectives of adaptation in IoT and CPS?	What changes in the environment can raise the necessity of adaptation?	This research question aims to identify and categorize the adaptation necessities due to changes in system, environment, and their coordination. These include changes in the environmental context and constraints, hardware layers, software components and connectors, and associated requirements.
		What changes in IoT/CPS HW/SW infrastructure can cause a need for adaptation?	
		How the dynamic coordination and interaction of IoT/CPS infrastructure and their environment can motivate adaptation?	
RQ2	What are the decision methods that can be adopted to realize adaptation in IoT/CPS?	What are the adaptation control times for IoT/CPS applications?	This research question focuses on IoT/CPS adaptation decision techniques, which imply control over IoT/CPS elements. The techniques can be categorized as model-based, rule-based, data-driven, optimization, or program-based, or a mix of them.
		What are the decision techniques that can be used as the substructure of IoT/CPS adaptation?	
RQ3	What kind of models, tools, or platforms are known by research and industry communities for IoT/CPS adaptation support?	Are the focus of adaptation supports on language or middleware levels, or domain-specific applications?	This question deals with various supports for IoT/CPS design and development. The subject attempts to discover existing platforms and applications, their features, and their requirements. The focus of this paper is especially on middleware support. Thus, the classified state of the art knowledge shall result in a set of middleware patterns potentially suitable for various domains.
		How can the adaptation support platforms satisfy industrial needs?	
		What range of application domains is addressed by each platform?	

whole systematic review methodology, with a notable influence on the primary studies search, the data extraction, and the data analysis processes.

It is worth mentioning that a good search strategy is expected to provide practical solutions to the following questions: *, which, where, what, and when* [18].

Which approaches? The search strategy consists of two phases: *i)* automatic search in scientific databases; and *ii)* snowballing. The first step will be performed using a search string (see below) based on identified keywords from research questions and areas of study. The search strings are used to retrieve potential primary studies through web search engines provided by digital libraries. Snowballing refers to using the reference list of a paper (backward snowballing) or the citations to the paper (forward snowballing) to identify additional papers [16]. The start set for the snowballing procedure is composed of the selected papers retrieved by the automatic search, namely the primary studies, which are selected by applying inclusion/exclusion criteria to the automatic search results. In any case, the inclusion/exclusion criteria will be applied to each paper. If a paper is considered to be included, snowballing will be applied iteratively, and the procedure ends when no new papers can be found.

(IoT OR "Internet of Things" OR IIoT OR CPS OR "cyber-physical" OR "cyber physical" OR cyberphysical OR pervasive) AND (middleware OR "middle-ware" OR "middle ware") AND ("self-adapt" OR "self adapt*" OR "self*" OR "adapt*" OR autonomic)*

Where to search? According to [18], it is essential to search for many different electronic sources because no single source can find all relevant primary studies. We followed the same procedure used for other systematic studies, such as [1, 2]. Table 4 shows the electronic databases that we will use for the automatic search as the primary source of literature for potentially relevant studies on the domain.

What to search? A suitable search string will be the input to the electronic data sources identified in the previous section, matching with paper titles, abstracts, and keywords. Following some test executions and refinements, the search string has been finalized, as shown above. We tried to codify the string in a way to be best adapted to specific syntax

Table 4: Electronic data sources targeted with search strings.

Library	Website
IEEE Xplore Digital Library	https://ieeexplore.ieee.org
ACM Digital Library	https://dl.acm.org
SpringerLink	https://link.springer.com
Web of Science	http://apps.webofknowledge.com
Wiley	http:/onlinelibrary.wiley.com
ScienceDirect	http://www.sciencedirect.com
Scopus	https://www.scopus.com

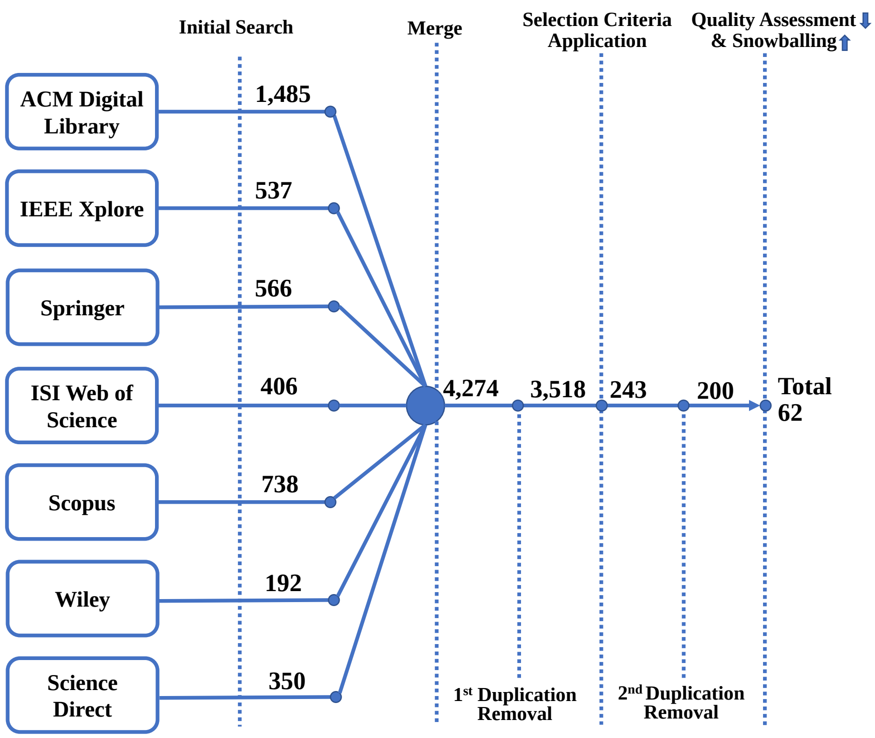

Figure 2: Search and selection process.

and criteria of each selected electronic data source. Further, we will combine all studies into a single dataset, after removal of impurities and duplicates.

When and what period to search? We do not consider publication year as a criterion for the search and selection steps. Thus, all studies coming from the selection steps, until June 2020, will be included regardless of their publication time.

3.2.2 Selection Strategy

A multi-stage selection process (Figure 2) has been designed to give full control of the number and characteristics of the studies coming from different stages [4]. As shown in Figure 2, we first applied the automatic search using the previously defined string on the electronic databases. This step resulted in 4,274 papers, which, after duplication removal, were reduced to 3,518. Researchers independently read the abstract of all studies selected and used the

[4]It is worth mentioning that on Springer, we considered "computer science" as the sub-discipline, and on Science-Direct, we searched on titles and abstracts only. These were due to avoiding a considerable number of false positives results.

inclusion and exclusion criteria (Table 5) to filter out irrelevant papers. A paper was included only when it satisfied all inclusion criteria and did not satisfy any exclusion criteria. The included papers of each researcher were checked by the others to minimize the bias.

Table 5: Inclusion and exclusion criteria.

Inclusion criteria	Exclusion criteria
Studies that propose modeling and/or analysis and/or development solution, architecture, method, and/or technique, specific for engineering (self-) adaptive middleware support for IoT/CPS.	Studies that, while focusing on IoT/CPS, do not explicitly deal with their (self-) adaptive middleware modeling and/or development aspects (e.g., studies focusing only on technological aspects and inner details of IoT/CPS).
Studies subject to peer review (e.g., journal papers, papers published as part of conference proceedings, workshop papers, and book chapters).	Secondary or tertiary studies (e.g., systematic literature reviews and surveys).
Studies written in the English language and available in full-text.	Studies in the form of tutorial papers or editorials. Because they do not provide enough information.

Applying the selection criteria led us to 243 studies. Although all the selected studies were on-topic, all three of us evaluated them qualitatively. The following quality assessment criteria were considered:

- **QA1)** What are the applicability and popularity of the research?
- **QA2)** Does the research contain novel and up-to-date methods and solutions?
- **QA3)** How can the research help design a set of adaptive middleware architectural patterns?
- **QA4)** Is the contribution well established and explained?
- **QA5)** Is the approach well evaluated?

The first quality assessment question evaluates if the method presented by a study is widely applied to other research or industrial cases. The second question rates the studies on the novelty of their problem-solving processes. The third question assesses the studies' architecture to see how it can support us in proposing architectural solutions. The fourth question analyzes the appropriateness of contribution and evaluation presented by each study. We calculated each study's total score by summing up the answer to each specific question $Q1 - Q4$ (Yes=1, Partly=0.5, No=0).

The quality assessment phase resulted in 59 studies, which increased to 62 by applying the snowballing process explained in the previous subsection. Among the reasons for which the snowballing added only a few primary studies, we bring up the effort we dedicated to design an inclusive search string and a careful selection that includes all significant studies on the topic.

After selecting a final set of primary studies, the data has been extracted to answer the research questions.

3.2.3 Data Extraction

This step aims to identify, collect, and classify data from the selected primary studies (the list will be available on an online data extraction file) to answer the research questions [5]. To this end, a detailed classification framework has been designed to structure the extracted data. Indeed, designing an effective classification framework needs a comprehensive analysis of the primary studies' content. Besides, the IoT/CPS standards and formal simulation and modeling classifications may support us through categorizing the data extraction. The systematic *keywording* process that we followed for this phase consists of collecting and clustering the keywords of primary studies.

- *Collect keywords and concepts*: researchers collect keywords and concepts by reading each primary study. When all primary studies have been analyzed, all keywords and concepts are combined to identify the research's context, nature, and contribution. The output of this stage is the set of keywords extracted from the primary studies.
- *Cluster keywords and concepts*: when keywords and concepts have been finalized, researchers can perform a clustering operation on them to have a set of representative clusters of keywords. This stage's output is the finalized classification framework containing all the identified attributes, each of them representing a specific aspect regarding (self-) adaptive middleware for IoT/CPS.

Table 6: Collected data items.

Data Item	Data Field	Research Question
DI1	Authors	Documentation
DI2	Year	Documentation
DI3	Title	Documentation
DI4	Venue	Documentation
DI5	Publication Trends	Documentation
DI5	IoT/CPS environmental context and constriants	RQ1
DI6	IoT/CPS hardware system	RQ1
DI7	IoT/CPS software components and connectors	RQ1
DI8	System goals, functional and non-functional requirements	RQ1
DI9	Coordination among CPS/IoT infrastructure and the environment	RQ1
DI10	Proactive and reactive adaptation	RQ2
DI11	Adaptation control and decision models	RQ2
DI12	Middleware support for IoT/CPS	RQ3
DI13	Application domain	RQ3
DI14	Industry adoption	RQ3

Furthermore, to have a rigorous data extraction process and ease the management of the extracted data, a structured data extraction form (in a replication package) will be designed. Once the data extraction form is set up, the researchers consider each primary study and fill the data extraction form accordingly.

3.2.4 Data Synthesis

The data synthesis activity involves collating and summarizing the data extracted from the primary studies [15] with the main goal of understanding, analyzing, and classifying current research on (self-) adaptation aspects of IoT/CPS middleware support. The data synthesis has been structured in two phases: vertical analysis and horizontal analysis.

- *Vertical analysis*: *i)* analysis of extracted data individually to track the trends and collect information of each study concerning the research questions; *ii)* analysis the discrete extracted data as a whole to reason about potential patterns and trends.
- *Horizontal analysis*: *i)* analysis of extracted data to explore possible relations across different dimensions and facets of the research. *ii)* using contingency tables analysis to cross-tabulate and group the data and made comparisons between two or more concepts of the classification framework.

3.3 Documenting

In the documenting phase, we report our main findings. The findings are usually clustered and classified into data items. Table 6 shows shows the data items [10] that are extracted to answer the identified research questions. Each specified data field my be divided into subcategories.

- **Documentation Data Items:** We extract authors, publication year, title, type, and venue of the chosen 62 primary studies. Figure 3 shows the distribution of adaptive IoT/CPS middleware support literature. It noticeably indicates that the number of papers grows by time, and 90% of papers are published within the last five years. This result confirms the recent scientific interest and research necessity on adaptive IoT/CPS middleware issues. The most common publication type is journal paper (30/62), followed by conference (23/62), workshop (7/62), and book chapter (2/62). Such a high number of journal and conference papers may point out that adaptive IoT/CPS middleware support is maturing as a research topic despite that it is still relatively young. Furthermore, we noticed that research on adaptive IoT/CPS middleware support is spread across many venues, mostly in the span of IoT (e.g., WF-IoT and IoTDI), control (e.g., CCTA), networking (e.g., NOMS), and computing (e.g., SOCA). The complete list of venues can be found in the data extraction file. However, the focus on the aspects mentioned above can prove the significance of distributed control and networking for adaptive IoT/CPS middleware design.
- **RQ1 Data Items:** An IoT/CPS should be adapted due to the changes that might happen in the environment, the infrastructure itself, and/or their coordination. An IoT/CPS is situated in the environment. The environment is the real world by which the software system might interacts. The environment might include both physical and virtual elements [19], that the system does not directly control its functionality. The system can

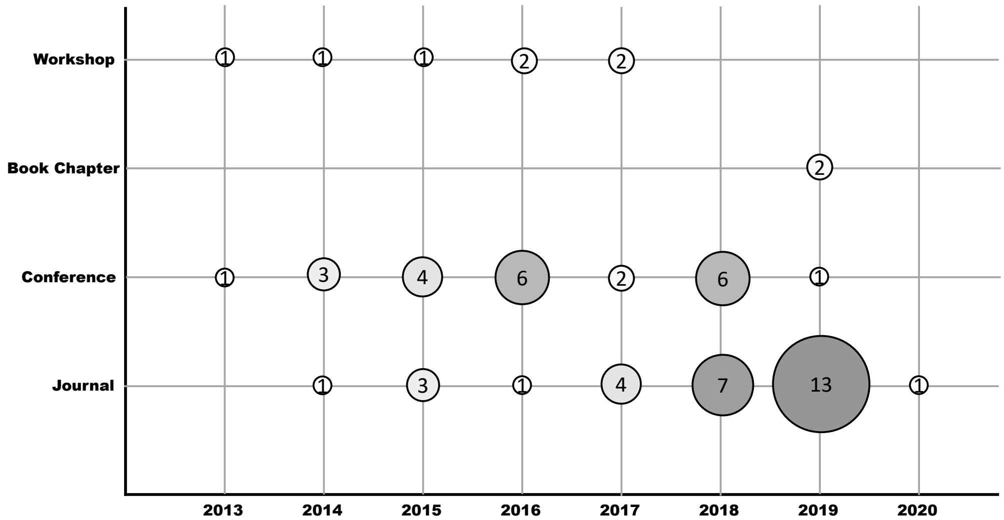

Figure 3: Primary studies distribution by publication type.

perform regardless of changes in the environment. However, most of the IoT/CPS systems interact with their environment in an ever-changing manner. The environment can be sensed and affected through *sensors* and *actuators*, respectively, which perform the functionality of IoT/CPS. Thus, the environmental context impacts the system goals and requirements and causes dynamic changes in the system.

IoT/CPS hardware architecture can be re-structured in run-time to add, delete, replace, migrate, and combine its elements. The IoT/CPS HW elements include sensors, network facilities, controllers, and actuators. The IoT/CPS software that is run on hardware elements includes a set of components bounded by connectors based on specific rules and constraints. IoT/CPS components include monitoring, processing, and execution software managed by functional and autonomic control loops. The control loops perform in a way to guarantee the functionality and quality of the system. They also respond to specific changes in goals and requirements. Such requirements can be both functional and non-functional and can necessitate changes in system and middleware architectures. The coordination of IoT/CPS and its environment should also be carefully analyzed and addressed by embedded systems, control of physical systems, and distributed systems concepts.

- **RQ2 Data Items:** The time aspect of IoT/CPS adaptation is related to when should the adaptation take place? The adaptation decision can follow a proactive or reactive strategy. If the IoT/CPS performs adaptation when a goal or requirement is already violated (e.g., a change in the resources or a drop in performance), it is reactive. If it adapts because of predicting any missed goals or requirements in the future, it is performing proactively. Users prefer proactive adaptation because of its ability to avoid quality degradation within the system. However, the proactive feature requires running complex prediction algorithms that depend on the correctness of entry data as parameters. Thus, a significant part of our primary studies focuses on reactive adaptation. In fact, the monitoring and execution activities are very much the same in reactive and proactive methods, but the analysis and planning phases make the difference.

  The decision method can be set based on various domains and fashions, such as model-based (e.g., model-predictive control, model-driven engineering, and agent-based modeling), rule-based (e.g., event-based, and reconfiguration rules), data-driven (e.g., machine learning, and reinforcement learning), optimization-based (e.g., cross-entropy), and program-based.

- **RQ3 Data Items:** The support to implement adaptive IoT/CPS could be in language, middleware levels, or specific domain requirements. The focus of this paper is on middleware support. We look into various middleware platforms that our primary studies use, specify if they are open-source, and assess their specifications and level of industrial adoption. Afterward, each middleware will be linked to application domains for which they are suitable. According to our study, some middleware proposals have the form of conceptual architectures. Some others customize their middleware and the rest design or use reliable middleware platforms that usually are open-source and can be reused.

After reporting the results, such as the one mentioned above, according to Peterson et al. [20], the quality rating for the systematic study will be assessed. Such a value is the ratio of the number of actions taken compared to the total number of actions reported in the quality checklist. However, the threats to validity are unavoidable. Below we shortly define the main threats to validity that we expect to face and mitigated during the research.

External validity: In our study, the most severe threat related to external validity may consist of having a set of primary studies that is not representative of the whole research on self-adaptive middleware support for IoT/CPS. We plan to mitigate this potential threat by *i)* following a search strategy, including both automatic search and backward-forward snowballing of selected studies, and *ii)* defining a set of inclusion and exclusion criteria. Along the same lines, gray and non-English literature are not included in our research. We want to focus exclusively on the state of the art presented in high-quality scientific studies in English.

Internal validity: It refers to the level of influence that extraneous variables may have on the design of the study. We plan to mitigate this potential threat to validity by *i)* rigorously defining and validating the structure of our study, *ii)* defining our classification framework by carefully following the keywording process, and *iii)* conducting a well-structured vertical analysis.

Construct validity: It concerns the validity of extracted data with respect to the research questions. We plan to mitigate this potential source of threats in different ways. *i)* performing an automatic search on various databases to avoid potential biases; *ii)* having a strong and tested search string; *iii)* complementing the automatic by the snowballing activity; and *iv)* rigorously screen the studies according to inclusion and exclusion criteria.

Conclusion validity: It concerns the relationship between the extracted data and the obtained results. We plan to mitigate potential threats to conclusion validity by applying well accepted systematic methods and processes throughout our study and documenting all of them in the excel package.

It is worth mentioning that we plan to report our main research-oriented findings and a detailed description of this study into an academic publication in a top-level academic journal.

4 Team

Three researchers carry out this study. Each researcher has a specific role within the team as follows:

- *Author 1*: A post-doctoral researcher with knowledge about IoT and the associated software architectures. Due to his experience in systematic reviews, he manages the majority of activities from planning the study to reporting.
- *Author 2*: A Researcher with experience in model-based control of autonomic, adaptive, and reconfigurable computing systems. In addition to performing various systematic study steps, he gives insights about the (self-) adaptation and interactive control loops.
- *Author 3*: An R&D engineer with several years of experience working on telecommunication, electrical system control, and cyber-physical systems. He supports conducting various steps, especially analyzing the selected primary studies and structuring the paper. He also develops the ideas on specifying industrial needs on adaptive IoT/CPS middleware support, suggesting novel tools and platforms, and providing industrial use-cases.

References

[1] Henry Muccini and Mahyar Tourchi Moghaddam. Iot architectural styles. In *European Conference on Software Architecture*, pages 68–85. Springer, 2018.

[2] Mahyar Tourchi Moghaddam and Henry Muccini. Fault-tolerant iot. In *International Workshop on Software Engineering for Resilient Systems*, pages 67–84. Springer, 2019.

[3] Henry Muccini, Romina Spalazzese, Mahyar T Moghaddam, and Mohammad Sharaf. Self-adaptive iot architectures: An emergency handling case study. In *Proceedings of the 12th European Conference on Software Architecture: Companion Proceedings*, pages 1–6, 2018.

[4] Danny Weyns. Software engineering of self-adaptive systems. In *Handbook of Software Engineering*, pages 399–443. Springer, 2019.

[5] Mirco Franzago, Davide Di Ruscio, Ivano Malavolta, and Henry Muccini. Protocol for a systematic mapping study on collaborative model-driven software engineering. *arXiv preprint arXiv:1611.02619*, 2016.

[6] Mirco Franzago, Davide Di Ruscio, Ivano Malavolta, and Henry Muccini. Collaborative model-driven software engineering: a classification framework and a research map. *IEEE Transactions on Software Engineering*, 44(12):1146–1175, 2017.

[7] Barbara Kitchenham, Rialette Pretorius, David Budgen, O Pearl Brereton, Mark Turner, Mahmood Niazi, and Stephen Linkman. Systematic literature reviews in software engineering–a tertiary study. *Information and software technology*, 52(8):792–805, 2010.

[8] Henry Muccini, Mohammad Sharaf, and Danny Weyns. Self-adaptation for cyber-physical systems: a systematic literature review. In *Proceedings of the 11th international symposium on software engineering for adaptive and self-managing systems*, pages 75–81, 2016.

[9] Muhammad Aufeef Chauhan, Muhammad Ali Babar, and Boualem Benatallah. Architecting cloud-enabled systems: a systematic survey of challenges and solutions. *Software: Practice and Experience*, 47(4):599–644, 2017.

[10] Stepan Shevtsov, Mihaly Berekmeri, Danny Weyns, and Martina Maggio. Control-theoretical software adaptation: A systematic literature review. *IEEE Transactions on Software Engineering*, 44(8):784–810, 2017.

[11] Ghazaleh Javadzadeh and Amir Masoud Rahmani. Fog computing applications in smart cities: A systematic survey. *Wireless Networks*, 26(2):1433–1457, 2020.

[12] Behrouz Pourghebleh and Vahideh Hayyolalam. A comprehensive and systematic review of the load balancing mechanisms in the internet of things. *Cluster Computing*, pages 1–21, 2019.

[13] Dat Dac Hoang and Hye-Young Paik Chae-Kyu Kim. Service-oriented middleware architectures for cyber-physical systems. 2011.

[14] Barbara Kitchenham and Pearl Brereton. A systematic review of systematic review process research in software engineering. *Information and software technology*, 55(12):2049–2075, 2013.

[15] Barbara Kitchenham and Stuart Charters. Guidelines for performing systematic literature reviews in software engineering. 2007.

[16] Claes Wohlin. Guidelines for snowballing in systematic literature studies and a replication in software engineering. In *Proceedings of the 18th international conference on evaluation and assessment in software engineering*, pages 1–10, 2014.

[17] Claes Wohlin, Per Runeson, Martin Höst, Magnus C Ohlsson, Björn Regnell, and Anders Wesslén. *Experimentation in software engineering*. Springer Science & Business Media, 2012.

[18] He Zhang, Muhammad Ali Babar, and Paolo Tell. Identifying relevant studies in software engineering. *Information and Software Technology*, 53(6):625–637, 2011.

[19] Danny Weyns. Software engineering of self-adaptive systems: an organised tour and future challenges. *Chapter in Handbook of Software Engineering*, 2017.

[20] Kai Petersen, Sairam Vakkalanka, and Ludwik Kuzniarz. Guidelines for conducting systematic mapping studies in software engineering: An update. *Information and Software Technology*, 64:1–18, 2015.

Primary Studies

- **P1**: Kit, Michal, Ilias Gerostathopoulos, Tomas Bures, Petr Hnetynka, and Frantisek Plasil. "An architecture framework for experimentations with self-adaptive cyber-physical systems." *In 2015 IEEE/ACM 10th International Symposium on Software Engineering for Adaptive and Self-Managing Systems,* pp. 93-96. IEEE, 2015.
- **P2**: Park, Soojin, and Sungyong Park. "A Cloud-based Middleware for Self-Adaptive IoT-Collaboration Services." *Sensors 19*, no. 20 (2019): 4559.
- **P3**: Souza, Alberto MC, and Jose RA Amazonas. "A novel smart home application using an internet of things middleware." *In Smart SysTech 2013; European Conference on Smart Objects, Systems and Technologies,* pp. 1-7. VDE, 2013.
- **P4**: Gascon-Samson, Julien, Mohammad Rafiuzzaman, and Karthik Pattabiraman. "Thingsjs: Towards a flexible and self-adaptable middleware for dynamic and heterogeneous iot environments." *In Proceedings of the 4th Workshop on Middleware and Applications for the Internet of Things,* pp. 11-16. 2017.
- **P5**: Mongiello, Marina, Tommaso Di Noia, Francesco Nocera, Eugenio Di Sciascio, and Angelo Parchitelli. "Context-aware design of reflective middleware in the internet of everything." *In Federation of International Conferences on Software Technologies: Applications and Foundations,* pp. 423-435. Springer, Cham, 2016.
- **P6**: Cecchinel, Cyril, François Fouquet, Sébastien Mosser, and Philippe Collet. "Leveraging live machine learning and deep sleep to support a self-adaptive efficient configuration of battery powered sensors." *Future Generation Computer Systems 92* (2019): 225-240.

- **P7**: Sylla, Adja Ndeye, Maxime Louvel, Eric Rutten, and Gwenaël Delaval. "Modular and hierarchical discrete control for applications and middleware deployment in iot and smart buildings." *In 2018 IEEE Conference on Control Technology and Applications (CCTA),* pp. 1472-1479. IEEE, 2018.
- **P8**: Banouar, Yassine, Saad Reddad, Codé Diop, Christophe Chassot, and Abdellah Zyane. "Monitoring solution for autonomic Middleware-level QoS management within IoT systems." *In 2015 IEEE/ACS 12th International Conference of Computer Systems and Applications (AICCSA),* pp. 1-8. IEEE, 2015.
- **P9**: Kim, Hyun-Woo, Jong Hyuk Park, and Young-Sik Jeong. "Efficient resource management scheme for storage processing in cloud infrastructure with internet of things." *Wireless Personal Communications 91,* no. 4 (2016): 1635-1651.
- **P10**: Handte, Marcus, Pedro José Marrón, and Gregor Schiele, eds. Adaptive Middleware for the Internet of Things: The GAMBAS Approach. *River Publishers,* 2019.
- **P11**: Lilis, Georgios, and Maher Kayal. "A secure and distributed message oriented middleware for smart building applications." *Automation in Construction 86* (2018): 163-175.
- **P12**: Ouedraogo, Clovis Anicet, Samir Medjiah, and Christophe Chassot. "A modular framework for dynamic qos management at the middleware level of the iot: Application to a onem2m compliant iot platform." *In 2018 IEEE International Conference on Communications (ICC),* pp. 1-7. IEEE, 2018.
- **P13**: Wu, Yu, and Minbo Li. "An IoT Middleware of Data Service." *In 2018 IEEE 11th Conference on Service-Oriented Computing and Applications (SOCA),* pp. 121-128. IEEE, 2018.
- **P14**: Mohalik, Swarup Kumar, Nanjangud C. Narendra, Ramamurthy Badrinath, Mahesh Babu Jayaraman, and Chakri Padala. "Dynamic semantic interoperability of control in IoT-based systems: Need for adaptive middleware." *In 2016 IEEE 3rd World Forum on Internet of Things (WF-IoT),* pp. 199-203. IEEE, 2016.
- **P15**: Wang, Feng, Liang Hu, Jin Zhou, and Kuo Zhao. "A data processing middleware based on SOA for the internet of things." *Journal of Sensors* 2015 (2015).
- **P16**: Kazmi, Aqeel, Zeeshan Jan, Achille Zappa, and Martin Serrano. "Overcoming the heterogeneity in the internet of things for smart cities." *In International workshop on interoperability and open-source solutions,* pp. 20-35. Springer, Cham, 2016.
- **P17**: Seiger, Ronny, Steffen Huber, and Thomas Schlegel. "Toward an execution system for self-healing workflows in cyber-physical systems." *Software & Systems Modeling 17*, no. 2 (2018): 551-572.
- **P18**: Palade, Andrei, Christian Cabrera, Fan Li, Gary White, Mohammad Abdur Razzaque, and Siobhán Clarke. "Middleware for Internet of Things: an evaluation in a small-scale IoT environment." *Journal of Reliable Intelligent Environments 4,* no. 1 (2018): 3-23.
- **P19**: Gomes, Berto de Tácio Pereira, Luiz Carlos Melo Muniz, Francisco José Da Silva e Silva, Davi Viana Dos Santos, Rafael Fernandes Lopes, Luciano Reis Coutinho, Felipe Oliveira Carvalho, and Markus Endler. "A middleware with comprehensive quality of context support for the internet of things applications." *Sensors 17,* no. 12 (2017): 2853.
- **P20**: Bao, Kaibin, Ingo Mauser, Sebastian Kochanneck, Huiwen Xu, and Hartmut Schmeck. "A microservice architecture for the intranet of things and energy in smart buildings." *In Proceedings of the 1st International Workshop on Mashups of Things and APIs,* pp. 1-6. 2016.
- **P21**: Ouedraogo, Clovis Anicet, Samir Medjiah, Christophe Chassot, and Khalil Drira. "Enhancing middleware-based IoT applications through run-time pluggable Qos management mechanisms. application to a oneM2M compliant IoT middleware." *Procedia computer science 130* (2018): 619-627.
- **P22**: Rahman, Mahmudur, Amatur Rahman, Hua-Jun Hong, Li-Wen Pan, Md Yusuf Sarwar Uddin, Nalini Venkatasubramanian, and Cheng-Hsin Hsu. "An adaptive IoT platform on budgeted 3g data plans." *Journal of Systems Architecture 97* (2019): 65-76.
- **P23**: Qin, Zhijing, Grit Denker, Carlo Giannelli, Paolo Bellavista, and Nalini Venkatasubramanian. "A software defined networking architecture for the internet-of-things." *In 2014 IEEE network operations and management symposium (NOMS),* pp. 1-9. IEEE, 2014.
- **P24**: Andersen, Michael P., Gabe Fierro, and David E. Culler. "Enabling synergy in iot: Platform to service and beyond." *Journal of Network and Computer Applications 81* (2017): 96-110.
- **P25**: Benson, Kyle E., Guoxi Wang, Nalini Venkatasubramanian, and Young-Jin Kim. "Ride: A resilient iot data exchange middleware leveraging sdn and edge cloud resources." *In 2018 IEEE/ACM Third International Conference on Internet-of-Things Design and Implementation (IoTDI),* pp. 72-83. IEEE, 2018.

- **P26**: Bahiri, M. N., A. Zyane, and A. Ghammaz. "An enhancement for the Autonomic Middleware-Level Scalability Management within IoT System using Cloud Computing." *In International Conference on Electronic Engineering and Renewable Energy,* pp. 80-88. Springer, Singapore, 2018.
- **P27**: Dobrescu, Radu, Daniel Merezeanu, and Stefan Mocanu. "Context-aware control and monitoring system with IoT and cloud support." *Computers and Electronics in Agriculture 160* (2019): 91-99.
- **P28**: Weißbach, Martin, Nguonly Taing, Markus Wutzler, Thomas Springer, Alexander Schill, and Siobhan Clarke. "Decentralized coordination of dynamic software updates in the Internet of Things." *In 2016 IEEE 3rd World Forum on Internet of Things (WF-IoT),* pp. 171-176. IEEE, 2016.
- **P29**: Naranjo, Paola G. Vinueza, Enzo Baccarelli, and Michele Scarpiniti. "Design and energy-efficient resource management of virtualized networked Fog architectures for the real-time support of IoT applications." *The journal of Supercomputing 74,* no. 6 (2018): 2470-2507.
- **P30**: Vanneste, Simon, Jens de Hoog, Thomas Huybrechts, Stig Bosmans, Reinout Eyckerman, Muddsair Sharif, Siegfried Mercelis, and Peter Hellinckx. "Distributed uniform streaming framework: an elastic fog computing platform for event stream processing and platform transparency." *Future Internet 11,* no. 7 (2019): 158.
- **P31**: Da, Keling, Philippe Roose, Marc Dalmau, Joseba Nevado, and Riadh Karchoud. "Kali2Much: a context middleware for autonomic adaptation-driven platform." *In Proceedings of the 1st ACM Workshop on Middleware for Context-Aware Applications in the IoT,* pp. 25-30. 2014.
- **P32**: Lalanda, Philippe, Stéphanie Chollet, Colin Aygalinc, and Eva Gerbert-Gaillard. "Service-based architecture and frameworks for pervasive health applications." *In 2015 IEEE 20th Conference on Emerging Technologies & Factory Automation (ETFA),* pp. 1-8. IEEE, 2015.
- **P33**: Qin, Zhijing, Luca Iannario, Carlo Giannelli, Paolo Bellavista, Grit Denker, and Nalini Venkatasubramanian. "MINA: A reflective middleware for managing dynamic multinetwork environments." *In 2014 IEEE Network Operations and Management Symposium (NOMS),* pp. 1-4. IEEE, 2014.
- **P34**: Caporuscio, Mauro, Vincenzo Grassi, Moreno Marzolla, and Raffaela Mirandola. "G o P rime: A Fully Decentralized Middleware for Utility-Aware Service Assembly." *IEEE Transactions on Software Engineering 42,* no. 2 (2015): 136-152.
- **P35**: Chen, Chunhua, and Junwei Yan. "HyTube: A Novel Middleware Layer for Smart Building Systems." In 2018 IEEE 16th Intl Conf on Dependable, Autonomic and Secure Computing, *16th Intl Conf on Pervasive Intelligence and Computing, 4th Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress* (DASC/PiCom/DataCom/CyberSciTech), pp. 135-142. IEEE, 2018.
- **P36**: Koutsandria, Georgia, Reinhard Gentz, Mahdi Jamei, Anna Scaglione, Sean Peisert, and Chuck McParland. "A real-time testbed environment for cyber-physical security on the power grid." *In Proceedings of the First ACM Workshop on Cyber-Physical Systems-Security and/or PrivaCy,* pp. 67-78. 2015.
- **P37**: Dutt, Nikil, Axel Jantsch, and Santanu Sarma. "Self-aware cyber-physical systems-on-chip." *In 2015 IEEE/ACM International Conference on Computer-Aided Design (ICCAD),* pp. 46-50. IEEE, 2015.
- **P38**: Siegemund, Gerry, and Volker Turau. "A self-stabilizing publish/subscribe middleware for iot applications." *ACM Transactions on Cyber-Physical Systems 2,* no. 2 (2018): 1-26.
- **P39**: Razouk, Wissam, Daniele Sgandurra, and Kouichi Sakurai. "A new security middleware architecture based on fog computing and cloud to support IoT constrained devices." *In Proceedings of the 1st International Conference on Internet of Things and Machine Learning,* pp. 1-8. 2017.
- **P40**: Singh, Jatinder, Thomas Pasquier, Jean Bacon, Julia Powles, Raluca Diaconu, and David Eyers. "Big ideas paper: Policy-driven middleware for a legally-compliant Internet of Things." *In Proceedings of the 17th International Middleware Conference,* pp. 1-15. 2016.
- **P41**: Prazeres, Cássio, Jurandir Barbosa, Leandro Andrade, and Martin Serrano. "Design and implementation of a message-service oriented middleware for fog of things platforms." *In Proceedings of the Symposium on Applied Computing,* pp. 1814-1819. 2017.
- **P42**: Merlino, Giovanni, Rustem Dautov, Salvatore Distefano, and Dario Bruneo. "Enabling Workload Engineering in Edge, Fog, and Cloud Computing through OpenStack-based Middleware." *ACM Transactions on Internet Technology (TOIT) 19*, no. 2 (2019): 1-22.
- **P43**: Rosa, Nelson, Gláucia Campos, and Davi Cavalcanti. "Using software architecture principles and lightweight formalisation to build adaptive middleware." *In Proceedings of the 16th Workshop on Adaptive and Reflective Middleware,* pp. 1-7. 2017.

- **P44**: Naber, Jens, Martin Pfannemüller, Janick Edinger, and Christian Becker. "PerFlow: configuring the information flow in a pervasive middleware via visual scripting." *In Proceedings of the 16th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services,* pp. 434-443. 2019.
- **P45**: Paspallis, Nearchos, and George A. Papadopoulos. "A pluggable middleware architecture for developing context-aware mobile applications." *Personal and Ubiquitous Computing 18,* no. 5 (2014): 1099-1116.
- **P46**: Portocarrero, Jesús MT, Flávia C. Delicato, Paulo F. Pires, Taniro C. Rodrigues, and Thais V. Batista. "SAMSON: self-adaptive middleware for wireless sensor networks." *In Proceedings of the 31st Annual ACM Symposium on Applied Computing,* pp. 1315-1322. 2016.
- **P47**: Javed, Asad, Jérémy Robert, Keijo Heljanko, and Kary Främling. "IoTEF: A Federated Edge-Cloud Architecture for Fault-Tolerant IoT Applications." *Journal of Grid Computing (2020)*: 1-24.
- **P48**: Baresi, Luciano, Sam Guinea, and Adnan Shahzada. "SeSaMe: towards a semantic self adaptive middleware for smart spaces." *In International Workshop on Engineering Multi-Agent Systems,* pp. 1-18. Springer, Berlin, Heidelberg, 2013.
- **P49**: Weyns, Danny. "Software engineering of self-adaptive systems." *In Handbook of Software Engineering,* pp. 399-443. Springer, Cham, 2019.
- **P50**: Bouloukakis, Georgios, Nikolaos Georgantas, Patient Ntumba, and Valerie Issarny. "Automated synthesis of mediators for middleware-layer protocol interoperability in the IoT." *Future Generation Computer Systems* 101 (2019): 1271-1294.
- **P51**: Benayache, Ayoub, Azeddine Bilami, Sami Barkat, Pascal Lorenz, and Hafnaoui Taleb. "MsM: A microservice middleware for smart WSN-based IoT application." *Journal of Network and Computer Applications 144* (2019): 138-154.
- **P52**: Rafique, Ansar, Dimitri Van Landuyt, Eddy Truyen, Vincent Reniers, and Wouter Joosen. "SCOPE: self-adaptive and policy-based data management middleware for federated clouds." *Journal of Internet Services and Applications 10*, no. 1 (2019): 2.
- **P53**: Pease, Sarogini Grace, Paul P. Conway, and Andrew A. West. "Hybrid ToF and RSSI real-time semantic tracking with an adaptive industrial internet of things architecture." *Journal of Network and Computer Applications 99* (2017): 98-109.
- **P54**: Colin, Aygalinc, Eva Gerbert-Gaillard, German Vega, Philippe Lalanda, and Stéphanie Chollet. "Autonomic service-oriented context for pervasive applications." *In 2016 IEEE International Conference on Services Computing (SCC)*, pp. 491-498. IEEE, 2016.
- **P55**: Donohoe, Michael, Brendan Jennings, and Sasitharan Balasubramaniam. "Context-awareness and the smart grid: Requirements and challenges." *Computer Networks 79* (2015): 263-282.
- **P56**: Zhao, Mengxuan, Gilles Privat, Eric Rutten, and Hassane Alla. "Discrete control for smart environments through a generic finite-state-models-based infrastructure." *In European Conference on Ambient Intelligence,* pp. 174-190. Springer, Cham, 2014.
- **P57**: de Brito, Mathias Santos, Saiful Hoque, Ronald Steinke, Alexander Willner, and Thomas Magedanz. "Application of the fog computing paradigm to smart factories and cyber-physical systems." *Transactions on Emerging Telecommunications Technologies 29*, no. 4 (2018): e3184.
- **P58**: Koziolek, Heiko, Andreas Burger, Marie Platenius-Mohr, Julius Rückert, Francisco Mendoza, and Roland Braun. "Automated industrial IoT-device integration using the OpenPnP reference architecture." *Software: Practice and Experience* (2019).
- **P59**: de Souza Cimino, Leonardo, José Estevão Eugênio de Resende, Lucas Henrique Moreira Silva, Samuel Queiroz Souza Rocha, Matheus de Oliveira Correia, Guilherme Souza Monteiro, Gabriel Natã de Souza Fernandes et al. "A middleware solution for integrating and exploring IoT and HPC capabilities." *Software: Practice and Experience 49*, no. 4 (2019): 584-616.
- **P60**: Seiger, Ronny, Steffen Huber, Peter Heisig, and Uwe Aßmann. "Toward a framework for self-adaptive workflows in cyber-physical systems." *Software & Systems Modeling 18*, no. 2 (2019): 1117-1134.
- **P61**: Casado-Vara, Roberto, Pablo Chamoso, Fernando De la Prieta, Javier Prieto, and Juan M. Corchado. "Non-linear adaptive closed-loop control system for improved efficiency in IoT-blockchain management." *Information Fusion 49* (2019): 227-239.
- **P62**: Portocarrero, Jesús MT, Flavia C. Delicato, Paulo F. Pires, Bruno Costa, Wei Li, Weisheng Si, and Albert Y. Zomaya. "Ramses: a new reference architecture for self-adaptive middleware in wireless sensor networks." *Ad Hoc Networks 55* (2017): 3-27.